09/242098

PCT/GB97/02159

The Patent Office

5

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP9 1RH

REC'D 01 OCT 1997
WIPO PCT

PRIORITY DOCUMENT

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed G D Court.

Dated 16 SEP 1997

Patents Form 1/77

Patents Act 1977
(Rule 16)

THE PATENT OFFICE
- 9 AUG 1996
RECEIVED BY HAND

The Patent Office

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form.)

Fee: £25

The Patent Office

Cardiff Road
Newport
Gwent NP9 1RH

1. Your reference — 37836/AJR

2. Patent application number *(The Patent Office will fill in this part)* — **9616755.6** — 09 AUG 1996

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

Mr Michael ~~J Kemp~~ Joseph Kemp
AJW 22/8/96
Casa Lucia
Vale Formosilho
S. Marcos da Serra
8375 S.B.de Messines
PORTUGAL

Patents ADP number *(if you know it)* — 7046030001

If the applicant is a corporate body, give the country/state of its incorporation

4. Title of the invention

AUDIO EFFECTS SYNTHESIZER WITH OR WITHOUT ANALYSER

5. Full name, address and postcode in the United Kingdom to which all correspondence relating to this form and translation should be sent

Reddie & Grose
16 Theobalds Road
LONDON
WC1X 8PL

Patents ADP number *(if you know it)* — 91001

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day/month/year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day/month/year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*

a) any applicant name in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

S PAGE BLANK (USPTO)

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	-
Description	14
Claim*(s)*	2
Abstract	-
Drawing*(s)*	16

10 If you are also filing any of the following, state how many against each item.

Priority documents	-
Translations of priority documents	-
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	-
Request for preliminary examination and search *(Patents Form 9/77)*	x
Request for substantive examination *(Patents Form 10/77)*	-
Any other documents *(please specify)*	-

11 I/We request the grant of a patent on the basis of this application.

Signature

Date

9 August 1996

12 Name and daytime telephone number of person to contact in the United Kingdom

AIDAN J. ROBSON

0171 - 242 0901

Warning

After an application for a patent has been filed the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or such direction has been revoked.

Notes

a) If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.

b) Write your answers in capital letters using black ink or you may type them.

c) If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.

d) If you have answered 'Yes' Patents Form 7/77 will need to be filed.

e) Once you have filled in the form you must remember to sign and date it.

f) For details of the fee and ways to pay please contact the Patent Office.

THIS PAGE BLANK (USPTO)

Audio Effects Synthesizer with or without Analyser

Introduction

In audio recording for music or film it is often desired to pass an audio signal through an effect unit to alter the sound in a desirable way, for example, in film work a recording may be made to sound as if it were coming through a telephone, from a distance or in a room with characteristic sound quality even though the original sound was recorded in a dead acoustic of a studio. In music work more severe distortions may be required, for example passing the signal through a guitar amplifier and speaker which is allowed to distort and back into a microphone, or through an analogue recording cycle onto and back from magnetic tape which is often considered to add a desirable sound quality.

Many devices exist to process signals in these ways, some specific to individual effects and some programmable to generate a range of effects on demand. The purpose of this invention to allow the simulation of a large variety of such effects and further to allow existing effects to be analysed and the characteristics of the effect to be stored and simulated on demand.

List of figures

The invention is described by means of reference to the attached figures which are described in detail after the following summary explanation.

Fig 1 shows the process of analysing an existing effect unit by means of applying an impulse and recording its impulse response.
Fig 2 shows the application of an input sound stream to generate a processed output stream by convolution with the sampled impulse response.
Fig 3 shows the application of impulses of different magnitudes to an effect unit to obtain more than one impulse response appropriate to different impulse amplitudes.
Fig 4 shows the application of an input stream to generate a processed output stream by modifying the convolution so that a different impulse repsonse may be applied to different input sample - in this case depending on amplitude of the input sample compared with a threshold shown dotted.
Fig 5 shows a further refinement where an input sample between two thresholds is applied proportionately to the two impulse responses appropriate to the thresholds on either side of the input sample.
Fig 6 shows an alternative step pulse that may be applied in the analysis process.
Fig 7 shows the derivation of the impulse response from the step response by means of a sample shift and a subtraction.
Fig 8 shows an arrangement of DSP and memory which can implement the steps of (i) analysing a device by means of generating impulses, storing the responses returned from an effect under analysis and performing various 'tidying up' algorithms as described below to create the stored impulse responses, (ii) reading an input sample and generating the sample, factor and address data for storage in memory as shown in fig 10, and (iii) executing the algorithm of fig12 to generate each output sample after each input sample has been read in, thresholded and stored. A fixed or removable disc

drive may also be provided for program storage, long-term storage of response data and exchange of data between machines.

Fig 9 shows part of the simulation processor wherein an input sample is analysed once to determine two impulse responses to be applied to it, where the start address of the impulse response stream in memory of the lower response appropriate to this sample is stored, and where the sample is divided proportionally as determined by the proximity of the sample amplitude to the two impulse response amplitudes and is for subsequent processing.

Fig 10 shows the algorithm of the applied to derive the values to be stored in fig 9.

Fig 11 shows the arrangement in memory after the most recent input sample has been divided and placed in memory at F1(0) and F2(0) together with the selected address of the lower of the two appropriate impulse responses stored at A(0. The previous samples derived values are stored at F1(1),F2(1), F1(2), F2(2) etc together with their associated A pointers for sufficient previous samples to exceed the length of the impulse responses used in the simulation.

Fig 12 shows the algorithm used to calculate an output sample from the data stored in memory in fig 11.

Fig 13 shows one possible multiple processor implementation wherein DSP1 is used first to analyse an effect and generate the sampled impulse responses, then is used during the simulation phase to generate the sample and factor memory entries. This memory is segmented into a number of areas each of which is accessible to its own DSP (2,3,4...) which can thus calculate part contributions to each output sample. These part sums are then fed back to DSP1 to be summed to generate the total output sample and fed to the output.

Fig 14 shows an alternative way to implement the simulation algorithm where the heavily repeated inner loop of the convolution algorithm is simplified for maximum speed of execution, requring a simple multiply of each element of the impulse response buffer and accumulation into each element of the output sample buffer.

Fig 15 shows a flow diagram of the process to generate the test pulses and record the impulse responses during analysis.

Fig 16 shows a flow diagram of an alternative process to generate the test pulses and record the impulse responses during analysis.

Fig 17 shows the test signal generated using the method of fig 16.

Fig 18 shows a noise removal strategy where impulse responses derived from lower amplitude impulses may be selectively replaced by impulse responses from higher amplitude impulses in areas where the mean amplitude of the impulse response falls below a threshold representing the approach to a noise floor which would impare the simulation process.

Analysis and Simulation of linear systems

The transfer characteristic of a linear audio processor can be characterised by its impulse response.A single pulse can be passed through an effect unit and the resulting signal which emerges can be recorded as a sequence of digital samples. The effect can then be simulated in the digital domain by convolving a digital input stream with this impulse response to produce a digital output stream which matches that which would have emerged from the sampled effect unit. The impulse response can be stored for recall later. This is illustrated in figure 1 where an impulse T is applied via a D/A converter 1 to produce an analogue impulse 2 which is fed into effect unit 3. The

output waveform 4 is fed via digital to analogue converter 5 and the resulting impulse response R is measured and stored.. Fig. 2 shows how the resulting impulse response R is used to calculate an output stream O from input stream I. Output sample O(0) is derived by taking input sample I(0) and multiplying this by the first sample of response R (R(0)shown at 7), summed (or *accumulated*) with the product of I(1) and the next older impulse sample (R(1) shown at 8) and so on until the oldest input sample required I(6) is multiplied by R(6) (shown at 10) is accumulated to make the latest output sample O(0). Thus the input stream of data I representing an input audio signal is convolved with the single impulse response R to produce each sample in output stream O. Although 6 samples are referred to here for the length of the impulse responses, this is for clarity only and in practice many more samples are used.

Where the effect unit to be analysed already has digital input and/or output the D/A (1) or the A/D (5) may not be required as the digital signals can simply be fed to or fed back from the effect unit..

Extension to non-linear systems

Many effects including some of those mentioned above are non linear in nature and the response of a signal path depends on the level of signal passing through the unit. To analyse such an effect unit it is possible to use a number of different impulses of different amplitude and to store a different resulting impulse response from each exciting impulse. This is illustrated in fig 3 for two different pulse amplitudes at fig 3(a) and fig 3(b). Figure 3(a) duplicates the process shown in fig 1, using a sample pulse T of maximum amplitude to determine the response of the system under maximum amplitude conditions. Figure 3(b) duplicates the test but using a lower amplitude impulse T', typically half the amplitude of the pulse in fig 3(a). The resulting impulse response is shown at R. This is then increased in amplitude to produce the response at R' by multiplying each sample by the ratio of the maximum sample amplitude at T over the lower sample amplitude at T'. This process is known as normalisation..

In practice to obtain a good analysis of the non-linear response of the system a number of different impulse levels are applied and a set of impulse responses (normalised to maximum amplitude) are obtained. Typically a set of 256 impulse responses are used using an equally spaced set of sample impulses from 1/256 of the maximum level up to the maximum level.

After obtaining the set of impulse responses it is possible to simulate the non-linear effect. When simulating the effect it is necessary to examine each input sample and depending on the magnitude of the sample to use the appropriate impulse response in the convolution. This is shown in figure 4 for the case where the set of impulse responses uses just the two responses obtained in fig 3 and by comparison with fig. 2. Each input sample (at I) needed to make up the output sample is compared against the threshold between the two sampled levels, shown as dotted line 11 representing the magnitude of the lower level sample pulse of figure 3b. If the input sample exceeds this threshold (i.e I(3), I(4) and I(5)), the impulse response of the higher amplitude pulse (shown at 12) is used in the convolution. If the magnitude of the input sample is

below the threshold (i.e. I(0). I(1). I(2). I(6)) the impulse response of the lower amplitude impulse (13) is used in the convolution calculation.

This process can be extended to use the impulse responses of any number of different impulse amplitudes by comparing the input sample against a number of thresholds. In the example where there are 256 equally spaced samples the level of the amplitude appropriate to any sample can be simply obtained by truncation of the magnitude of the sample to 8 bits (equivalent to 256 levels). The magnitude means that the sign of the sample value is removed to determine solely its amplitude.

In fact it can be seen that the number of calculations required to generate an output sample is increased only by the need to make a decision for each input sample. The decision needs only to be taken once for each input sample (regardless of how many times this sample needs to be used to calculate subsequent output samples) so in fact represents only a small increase in computational complexity. This is shown in the later detailed description of the process of simulation. This is important because it is possible to use a large number of different impulse responses representing, say, 256 different sample levels without increasing the number of calculations by anything like the number of levels used.

Improvement by linear interpolation of impulse responses

Whilst the above process provides a simulation of the sampled effect, an improvement in distortion characteristics can be made if desired at the expense of some increase in computational complexity by modifying the process so that instead of selecting between two different impulse responses at a given level, a cross-fade effect is used applying a proportion of the input sample to two impulse responses representing two adjacent sample levels. This is shown in fig 5 where a sample (14) a quarter of the way between two sample thresholds (15, 16) is applied three-quarters to the impulse response representing the lower sample level (17) and one quarter to the impulse respomse representing the higher sample level (18). No calculation needs to be performed with any of the other impulse responses. The computational complexity has thus doubled over the simple case of fig 2 plus the additional compution to compute the ratio between the two levels. Although this represents more complexity than of the simple case of fig 2, it still represents an acceptable level of complexity to achieve the non-linear characteristic of many simulated effects, as once again this can be evaluated just once for each input sample.

Switching between modes

In fact the simulator can be made to switch between the three cases of the simple linear simulator of fig 2. the non-linear simulator of fig 4 and the improved non-linear simulator of fig 5 according to the available computational power and according the length of the impulse responses used. This switching can be achieved by changing the stored program executed by the DSP processors used to implement the system.

Reducing Noise in the sampled impulse response using an alternative sampling pulse

The analysis pulse of fig 1 generates an impulse response but the resulting impulse response may also contain noise. Low frequency noise tends to be correlated between adjacent samples and during the resulting simulation phase may lead to either large DC offsets or general low frequency noise on the resulting output.

Figure 6 shows that instead of the unit impulse test signal T of figure 1 the step pulse ST may be applied. The step response SR is thus obtained.

Figure 7 shows how to recover the unit impulse response required R. The step impulse response SR is shifted on by one sample to get SR' which is subtracted sample by sample from the response SR to yield the desired impulse response R. Thus any substantial correlation between samples is largely eliminated, and any DC offset (i.e. a constant bias found on all analysed samples) is totally removed. This can of course be calculated as the impulse response is sampled by storing the previous sample value S_{n-1} amd subtracting it from the current sample S_n so the value $S_{n-1} - S_n$ is stored as the desired impulse response.

The desired response at a number of different amplitudes can be found by using steps of a number of different sizes.

Implementing the Analysis and Simulation

The implementation of the analysis and simulating machine will now be described by reference to figures 8 to 17. Figure 8 shows one arrangement using a stored program computer optimised for digital signal processing. Typically one or more digital signal processor (DSP) devices 21 are used. The DSP is attached to memory for impulse responses 22 and for digital audio sample and control data 23, as well as program memory 24. These may in fact be part of a single general purpose memory array or for example the program memory 24 may be part of a separate array for higher performance. Audio input is provided either via analogue to digital converter 25 or via direct digital input 26 and audio output is fed via digital to analogue converter 27 and via a direct digital output 28. A disk storage subsystem 28 is also connected and a user control panel and display 30 is provided to allow the user to initiate analysis, store or select stored impulse responses, and select simulation modes, as well as initiate editing of impulse responses as described later

This device may generate the analysis pulses, store and process the resultant impulse responses, and produce the simulation by loading the appropriate control program from disk or other storage medium

One method of implementing the process of simulation will be described first.

Figure 9 shows the process of reading in samples to be processed. A number of impulse responses are stored in arrays of memories shown at 31, 32 and 33. Three are shown but in practice any number may be used. These are identified by the address in memory of the first element of each response at 34, 35 and 36 and the memory array A can store these memory addresses, or pointers, represented by the arrows shown from memory elements of array A pointing to the appropriate impulse array. Each input sample arriving has one element of A reserved for it to denote the appropriate pair of

impulse responses 31 - 33. The pointer addresses the lower of the two impulse responses representing the threshold below the input sample, and the second impulse response is always the next one above representing the next higher threshold level of the input sample.

Memory arrays F1 and F2 store a pair of factors which are derived from the input sample and represent the input sample divided into two parts, one of which will be applied to the lower impulse response and one of which will be applied to the higher impulse response. The sum of these two factors is always the input sample value itself and the sample is divided and stored in elements of arrays F1 and F2 according to the proportion to be applied to each impulse response. Each input sample 37 therefore is divided in process 38 and loaded into the next free set of elements of the arrays A, F1, F2. A pointer 39 is then incremented to point into the next set of elements for the next input sample when it arrives.

Figure 10 shows by means of a flow diagram the details of the process 38. The input sample is compared with the various thresholds T_1, T_2 etc representing the levels at which the impulse responses were sampled to find the two thresholds T_n and T_{n+1} below and above the input sample amplitude, i.e. such that

$T_n \leq |S| < T_{n+1}$.

It should be noted that if the number of equally spaced levels is a power of 2 (e.g. 256) the threshold value T_n can be determined by first removing the sign of the sample value then truncation to the number of bits appropriate to the power of 2, (e.g. 8).

The next step is to calculate the proportion by which the sample amplitude exceeds the threshold (shown as factor k), then divide the sample in this proportion to place into arrays F1 and F2.

The input pointer is then advanced ready for the next sample. The array stored will be used for calculating each output samples up to the length of the impulse responses, so after a number of output samples (for example 5000) the values just calculated will no longer be required. Standard techniques may be applied to implement a 'cicular buffer' where the pointer can be wrapped back to the start after this many samples, thus limiting the size of the arrays. These techniques are well known and do not need to be described further here.

Figure 11 thus shows the layout of data in memory after a number of samples have been read in and processed to calculate output samples (ignoring any issues relating to circular buffers). In this example and in the process shown in fig 12 the parenthesised suffix (0) is used to indicate a value relating to the most recent sample, (1) the next older and so on. For the impulse response the suffix (0) means the first sample in the impulse response buffer (i.e. the first that arrived during the analysis process), (1) the next older etc up to (M-1) which represent the most delayed impulse response sample, where M is the number of samples in each impulse response buffer.

Figure 12 shows the flow diagram to calculate each output sample. It comprises a main loop starting at 43 which is executed M times for each output sample by means of the

control variable J which is zeroed at 41. The output sample is accumulated into the variable SOUT and so this is zeroed at 42 before entering the loop. The first step in the loop at 43 (for the element J) is to load the impulse response pointer A(J) (being the Jth element of array A). Using this pointer it is possible to load the appropriate impulse response sample from each of the appropriate response arrays. These are referred to as I1 read from A(J)+J (at step 44) and I2 read from A(J)+J+M (at step 45).

The two parts of the input sample F1 and F2 may be read from the F1, F2 arrays at offset J at step 46. The two multiply and accumulate steps can be performed to accumulate the output sample into SOUT as shown at step 47. It is then only necessary to increment J (at step 48) and to test this against M (at step 49). When J reaches M the output sample is complete and the loop is finished.

The output sample value may then be fed to the output of the machine (fig 8 items 27 and 28). Output pointers can then be moved on one sample ready for the next output sample.

It should be mentioned that if either of the two simplified processes of fig 2 or fig 4 is to be carried out some simplification of the above processes can be employed. For example, if cross-fading is not to be employed (as described in figure 4) the sample is not divided between F1 and F2 but is simply stored wholely in array F1. Thus all F2 values are considered to be zero and so the memory array is no longer required and any steps relating to F2 can be by-passed, i.e. step 45, the second read at step 46 and the second multiply accumulate at 47. If the basic linear simulation of fig 2 is required the process is further simplified to eliminate thresholding the input sample, it is simply stored in array F1. Only one address of impulse response is needed so array A is now no longer required and in figure 12 step 43 is not required and the sole base address of the one impulse response is used instead of A(J) at step 44.

It will be appreciated that the number of operations can be substantial as the length of the impulse responses used (M) may typically be 5,000 or longer. Accordingly, and depending on the speed of the DSPs it may be necessary to use more than one DSP to operate in real-time.

Figure 13 shows one possible architecture of such a machine. DSP 51 processes the input sample into the arrays A, F1 and F2 as already described but which are stored in segmented memory arrays 52. This memory is arranged so that it can be wholly accessed by DSP51 for loading with processed input samples, but is separated into sections which can be individually accessed by DSPs 53, 54, 55 etc. Each DSP thus has access to part of each array and for each output sample can perform part of the multiply accumulate loop described in figure 12. The resulting parts of the accumulated output sample are written back to more shared memory 56. DSP 51 (which otherwise is not heavily occupied by the input process) then adds all the separate parts together to produce the whole output sample. Thus ten processors (53, 54 etc) could be used so that each performs 500 accumulation steps per output sample, and DSP 51 then has to sum the 10 partial values. Thus 5,000 step impulse responses may be subdivided as appropriate to the speed of the DSP processors. Each DSP 53, 54 etc is effectively executing the same program and so may be fed from either the

same or separate program memories 57, 58 etc. It is only necessary to map each part of the memory 52 to appear at the same address location in each associated DSP.

It should be mentioned that there are other ways of dividing up the process which is functionally identical, producing identical output for the same data. For example fig 14 shows a rearrangement of the process so that the bulk of the processing is done for each input sample, accumulating output as the input samples appear. After the Mth input sample is accumulated into the output sample buffer the first output sample is ready for output. Thereafter after each input sample is accumulated, another output sample us available.This arrangement may suit some DSP architectures better depending on the exact nature of the DSP's instruction set.

Figure 15 shows the process for generating the analysis pulse and reading the impulse response data. If it is only desired to obtain one analysis at one amplitude it is not necessary to go round the loop more than once, and a single amplitude should be used that does not overload the unit under test (refer to the configuration of figure 1). To determine this amplitude may require some repeats of the process and auditioning of the simulation as well as observing any overload indication on the unit under test.

To obtain a set of impulse responses it is advisable to start with the lowest amplitude as shown at step 61 of figure 15. At step 62 it is desirable to wait for any residual effect of a previous signal passing through the unit under test, as some effect devices may continue to generate output for some time after stimulation, for example due to resonances or reverberations. This process is done by monitoring the return signal from the device under test and observing the noise floor. If this is decaying over a short space of time the process simply waits for the noise floor to become stable. It would generally be advisable to apply to time limit which can be user selectable in case of a varying noise floor causing an indefinite wait.

At step 63 a test pulse of the desired amplitude is emitted by setting the output stream to the value A in one sample period and back to zero at the following sample. At step 64 the returning stream is monitored and stored (usually into RAM) until the time limit set by the machine is reached. This is determined by the number of steps which the simulator can process in real time, or can be limited by memory available or be further limited by user intervention to minimise processing requirements. It should also be noted that the process of step 62 can also be followed to determine when there is no significant further response and further used to shorten the sampling process.

If the amplitude of the test pulse is less than the maximum amplitude then each element of the measured response is scaled up in the proportion $A_{max}/A_{current}$ to normalise the response stored at step 64.

Once the sampling is complete the amplitude is tested at 65 to determine if the process is complete. If not, the next higher level of amplitude can be loaded into A and the loop repeated.

It is possible to use a descending set of test pulses instead but the two advantages of an increasing stream is (i) there is likely to be less ongoing disturbance from preceeding test pulses as they were of lower amplitude, and (ii) in the event that the operator hears

the increasing pulse levels causing damage to equipment (or his hearing) he can intervene to stop the test.

A useful refinement is to allow the system to generate a continuous stream of pulses at user definable amplitudes solely for the purpose of allowing the operator to select the optimum levels of signal to pass through the device under test.

Figure 16 shows the improved method of sampling the test unit and figure 17 shows the resulting test signal applied to the D/A converter and digital output of the machine during the test process outlined in fig 6. This is described by reference to both fig 16 and 17.

Once again the process can be done just once using a single amplitude test pulse or the loop shown in the diagram can be carried out to obtain the set of responses.

At step 71 the initial minimum amplitude value is selected. The output stream fom the pulse generator is set to the value $-A/2$ at step 72, producing the output step 81 in fig 17. At step 73 it is necessary to wait for any resulting effects from the unit under test to die out. In this case the difference signal from the device under test is used. This is specified as the current sample received S_n minus the previous sample S_{n-1}. This stream is measured to determine when the device under test reaches the general noise floor of the device.

The test signal is now generated by stepping the output stream by the amplitude A, by stepping in a direction to cross the zero value, as described at step 74. This is shown at 82 in fig 17 for this first value. The resulting output of the device under test is now monitored and the difference stream $S_n - S_{n-1}$ is stored as the impulse response.

If the amplitude of the test pulse is less than the maximum amplitude then each element of the measured response is scaled up in the proportion $A_{max}/A_{current}$ to normalise the response stored at step 75.

At step 76, value A is tested to see if it has reached the maximum step desired and if not it is increased to the next amplitude to test (step 77). The process then loops back to step 73 where the effect of the stimulation is allowed to die out, then the output is stepped again, this time in the opposite direction. This is shown at 83 in fig 17. Once again the output stream is monitored and the difference signal is stored. It is not shown in the diagram but when the test pulse is negative-going the resulting impulse response has its sign reversed when being stored to compensate for the opposite sense of the test pulse.

When the complete set of impulse responses has been stored they may be windowed, edited in duration or subjected to noise improvement processing as described later.

It should be noted that the step of waiting for any residual stimulation of the device under test (shown at step 62 of fig 15 and step 73 of fig 16) may be replaced by a fixed wait period in many instances where there is not significant energy storage in the device under test. This has a benefit that the output test signal becomes the same for any test and may thus be recorded (preferably in digital format) for application to a

device remote from the analysis machine. The resulting output from the device under test may also be recorded and can later be analysed by the analysis process. The only significant alteration to the process of analysis is that instead of generating the test pulse it is necessary simply to wait for any significant response to appear in the recorded stream and store this and the following impulse response as being the response to the first test pulse, then similarly wait for the appearance of further responses to later pulses and thus obtain a complete set of impulse responses. This is useful as an operator may simply carry a tape of the test stream and if he encounters a device which he wishes to analyse he simply plays the tape through the device and records the result for later analysis and simulation. A useful refinement to improve this process is to precede the test signal with a short burst of tone which both can be used for level setting and can be recognised at the analysis stage and taken as a trigger to start the process of looking for response signals at a known period after the tone burst.

Although the sampled effect is shown as an analogue device, a digital processor may be sampled by applying the sample impulse directly to the digital input and sampling directly the output impulse response.

Improving noise

A potential problem with the system is that significant noise generated by the device under test will appear as noise in the simulated effect. This can be made worse when using impulse responses derived at low levels of test. However since many effects become linear as the level through the device decreases it is often just necessary to use a set of impulse responses derived at relatively high levels, and below this threshold of linearity, to use the impulse response derived at the highest linear level in place of all lower impulse responses. This can be done under manual intervention from the operator who can choose a balance between desirable non-linearity and accepable noise by auditioning the effect of selective replacement.

Where it is not possible to achieve a desirable balance because it is desired to preserve lower level non-linearities where noise is a problem, it is possible to selectively modify parts of the impulse responses derived at low levels by replacement with matching parts of the responses from higher level impulse responses, where the areas to be replaced are determined by evaluating the absolute amplitude of each section of the response and replacing it where the impulse response is seen to be near the noise floor.

Figure 18 shows some details of this process. A higher level impulse response is shown at (a) and a lower level one at (b). An envelope 91 (at (c)) is generated representing the average level of a local region of the impulse response (b). This is evaluated by calculating the RMS value of the nearby samples, weighted towards the current time for each point in the envelope. In practise this may encompass 400 to 500 samples. The envelope 91 is compared with a threshold 92 which may be user determined or estimated by comparing with the noise floor found during the analysis processes of fig 15 and 16. In fig 18 (c) it can be seen that the example envelope 91 falls below the threshold at 93 and rises again above it at 94. From this a 'cross-fade' envelope is generated (d). Using this envelope the impulse response (b) is selectively replaced with impulse response (a) with a soft crossfade of several milliseconds at each end of the replacement (shown as the ramps 95 of the crossfade envelope) to generate a

new impulse response (c) where the lower level area is replaced by the lower noise floor impulse response taken at the higher level (a).

The new impulse response is generated according to the formula

r=c.a + (1-c).b

where c is the envelope value, a is the sample value from the higher level impulse response and b is the sample value from the lower level impulse, and r is the resultant sample to replace in the lower level sample. The period (.) represents multiplication. This provides a crossfade to the higher level impulse response where the lower level signal was below the threshold.

To determine the noise floor automatically it will be seen that for the impulse responses taken at lower levels there will be a level which the envelope never drops below due to noise. The threshold can thus be set say 50% above this and applied progressively from the near the highest level sample down to the lowest level. It is appropriate to start the process at the impulse response some 12dB below below the maximum, in other words that sampled with a sample pulse about a quarter of the amplitude of the highest sample impulse used.

Length of impulse responses and processing power

The impulse response lengths required depend on the energy storage characteristics of the effect sampled. Typically an equaliser, valve amplifier or speaker/microphone combinations in short reverberation environments can be simulated with impulse times of up to 1/10th second, or for example 5,000 samples. Each output sample will require the accumulation of 5,000 values of input sample multiplied with 5,000 impulse response samples, or 250 million operations per second assuming a 50,000 sample per second sampling rate. Thus the simple case of fig 2 requires 250 million multiply accumulates (MAC) operations on linear arrays of data, while the process shown in fig 12 requires correspondingly more steps to be repeated this many times.

To simulate fully reverberant effects, impulse responses of several seconds can be needed resulting in a proportional increase in processing power. This is quite possible within a network of DSP chips. To make the best use of a particular hardware implementation however the simulator should be arranged to switch amongst the three simulation methods described: the linear simulation of fig 2, the simple non-linear simulation of fig 4 and the interpolated simulation of fig 5 (shown in greater detail in figs 8 onwards). This means that in simulations where non-linearity is not required more processing power is available for longer impulse responses and therefore longer reverberent periods.

Windowing of impulse responses

It should be noted that where an effect is sampled but the impulse response exceeds the length of sample which it is possible to calculate in real-time in a particular hardware implementation, it is necessary to truncate the impulse response by windowing the response, i.e. effectively fading off the last 1/20th second or so linearly to zero. In fact

all sampled impulse responses should be windowed in this way to prevent any glitch effects from suddenly truncated noise signals.

Editing impulse responses

Trimming the start and end

There is always some delay between the application of an impulse to a device and the output response. This results in an equal delay in the simulation. Sometimes the effect can be improved by removing or reducing this delay and in any event this shortens the sample to reduce compuational requirement. It is simple to arrange for the operator to trim off samples from the front of the sample - the effect of which he can audition to his taste, or a threshold level can be set on a response to automatically trim off any initial response below this 'noise' threshold. This threshold would typically be applied to the impulse response derived from the highest level sampled signal and once determined, the same amount is trimmed off the start of the whole set of impulse responses.

Frequency shifting

Interesting variations of the sampled effect may be made by resampling each impulse response to a higher or lower frequency using standard resampling algorithms. The effect of each change can be auditioned to the taste of the operator. This allows various effects, such as for example the resonances in the sampled effect being matched to dominant frequencies in the signals to be processed.

Combination of effects

It is possible to simulate the effect of passing a signal through two successive effects by taking each impulse response of the first effect and passing it through the simulation of the second effect to generate a new impulse response for that sample amplitude. This is done for each impulse response of the first effect to achieve the same number of new impulse responses representing the combined effect. In the case of the simple method of fig 2 this represents s simple convolution of the impulse responses.

Interpolation and extrapolation effects

The set of impulse responses representing the range of levels passing through an effect embodies the non-linear characteristic of the sampled effect. New and interesting effects can be achieved by partially linearising the effect. To do this a subset representing a range of the original set is taken and a new complete set of impulse responses is generated by interpolation of each sample step through the set of impulse responses.

It is also possible to make the non-linearity more extreme by extrapolating beyond the original range. This can result on extreme values of samples and generally the whole sample set will have to be attenuated to keep the output within acceptable limits.

After any such recalculation the operator can again audition the effect to achieve a desired effect. The extrapolation effects will generally become very strange but small amounts of extrapolation may generate desirable distortions.

Deriving impulse responses from virtual systems

It should be noted that as well as sampling existiing effects it is quite possible to generate a computer model of a new device and calculate a set of impulse responses. These may then be loaded into the simulator to allow the effect to be auditioned in real-time. In this way the simulator can emulate arbitrary digital effects such as equalisers, or simulated physical models e.g. room simulations, and especially non-linear devices such as amplifier or loudspeaker simulations.

In the case of simple equalisers which are linear in character only one impulse response is generated for any chosen equaliser. These can be calculated and loaded rapidly to allow real-time variation of equaliser characteristics. The simulator thus provides a powerful simulator of a wide range of equaliser devices complete with real-time user control of parameters. In practice when a parameter is varied the new impulse response is calculated and loaded and a cross-fade can be performed to the new effect to remove switching effects when parameters are varied.

Arithmetic

As with all good signal processing practise care must be taken with rounding or truncation of digital value. It is best to preserve precions of all calculations to for example 32-bits if fixed point arithmetic is used of 24-bits of mantissa if floating point is used. Final digital output can be reduced to the desired digital output format using appropriate and known bit reduction techniques.

Further Uses

The process described can be used to simulate effect which are asymmetric by also taking into account the sign of the signal to be processed and taking separate analysis samples for positive going test pulses and negative going test pulses. This asymmetric processing could be appropriate, for example, to simulation of high sound pressure level effects in air where the sound carrying capacity of air is asymmetric.

A further use of the process of selecting between impulse responses is for using some other characterisitic than the amplitude of the incoming sample to control selection. For example a number of different effects can be placed into each impulse response memory and be selected between (including using the crossfading techique) under user control or in a repetitive manner using a control oscillator. In this way a time varing effect can be simulated, for example a rotating Leslie loudspeaker cabinet or a varying flanger or phaser effect. The required impulse responses can either be calculated to generate an effect or an existing unit can be sampled at a number of different settings representing a range which the effect is normally used to sweep through. Thus a Leslie loudspeaker can be analysed at a number of different static positions of the rotating speaker and the resulting set of impulse responses stored. Then cycling through the responses will simulate rotation of the speaker (including the doppler effects of the

moving speaker as different impulses responses will have different delays built in representing different direct and indirect signal paths from the loudpseaker analysed).

Although a mono system is described typically two units will run in parallel to allow stereo in and stereo out. Often the input signal will be the same applied to both channels to generate stereo samples effects from mono sources.

Non real time and general purpose computers

Note that it is also possible to process in non-real time using less hardware and this can be done on typical general purpose desk-top computers. However the best use is achieved when operating in real-time whether this is on a high performance general purpose computer implementing the algorithms described or by means of dedicated multiple DSP architectures.

CLAIMS

1. A method for simulating the effect of an audio effect processor comprising the steps of:

a) storing the impulse response of the audio processor; and

b) applying the stored impulse response to an input signal to derive an output signal to which the audio effect of the processor has been applied.

2. A method according to claim 1, wherein the step of storing the impulse response comprises storing a set of digital samples representing the impulse response and the step of applying the stored impulse response comprises the step of convolving a first series of digital samples representing the input signal with the set of digital samples representing the impulse response and to give a second series of digital samples representing the output signal.

3. A method according to claim 1 or 2, including the step of storing impulse responses for at least two impulses of different amplitudes and the step of applying the impulse response to the input signal comprises the step of assessing a characteristic of the input signal and selecting one of the impulse responses to apply to the input signal in dependence on the result of the assessment.

4. A method according to claim 3, in which the step of assessing a characteristic comprises determining whether

the amplitude of the input signal is above or below a predetermined threshold.

5. A method according to claim 4, further comprising the step of determining whether or not the amplitude of the input signal falls within a predetermined range, applying more than one impulse response to the input signal if the result of the determination is that the amplitude of the input signal does fall within the predetermined range and deriving the output signal therefrom.

6. A method according to claim 5, in which the more than one impulse responses applied to the input signal are applied in proportions which sum substantially to 1.

7. A method according to claim 6, in which the proportions of the impulse responses applied to the input signal are dependent on the position of the amplitude of the input signal within the predetermined range.

Fig 1

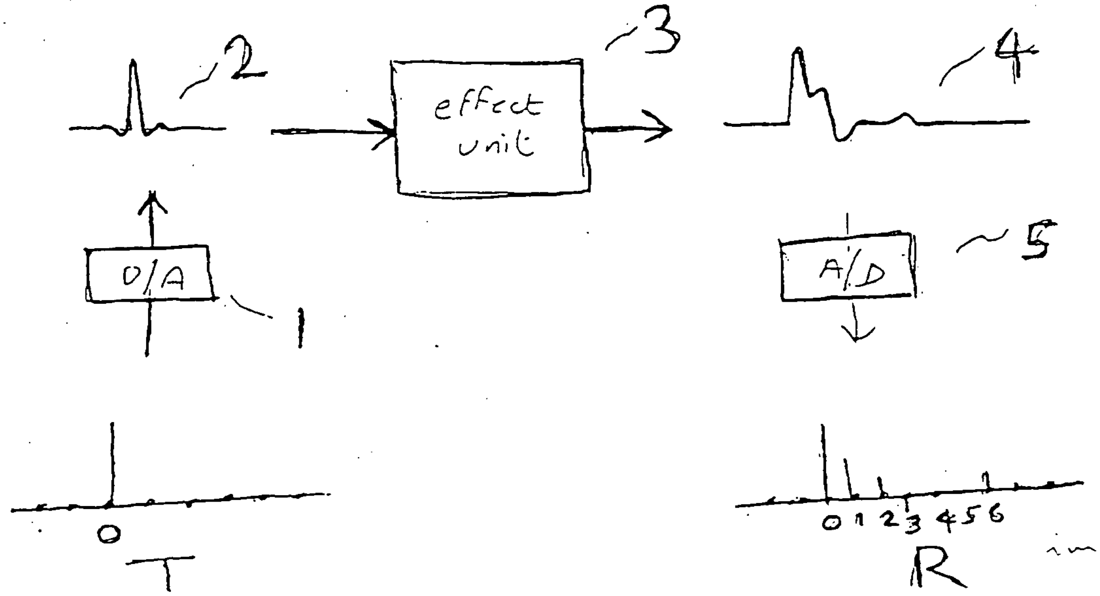
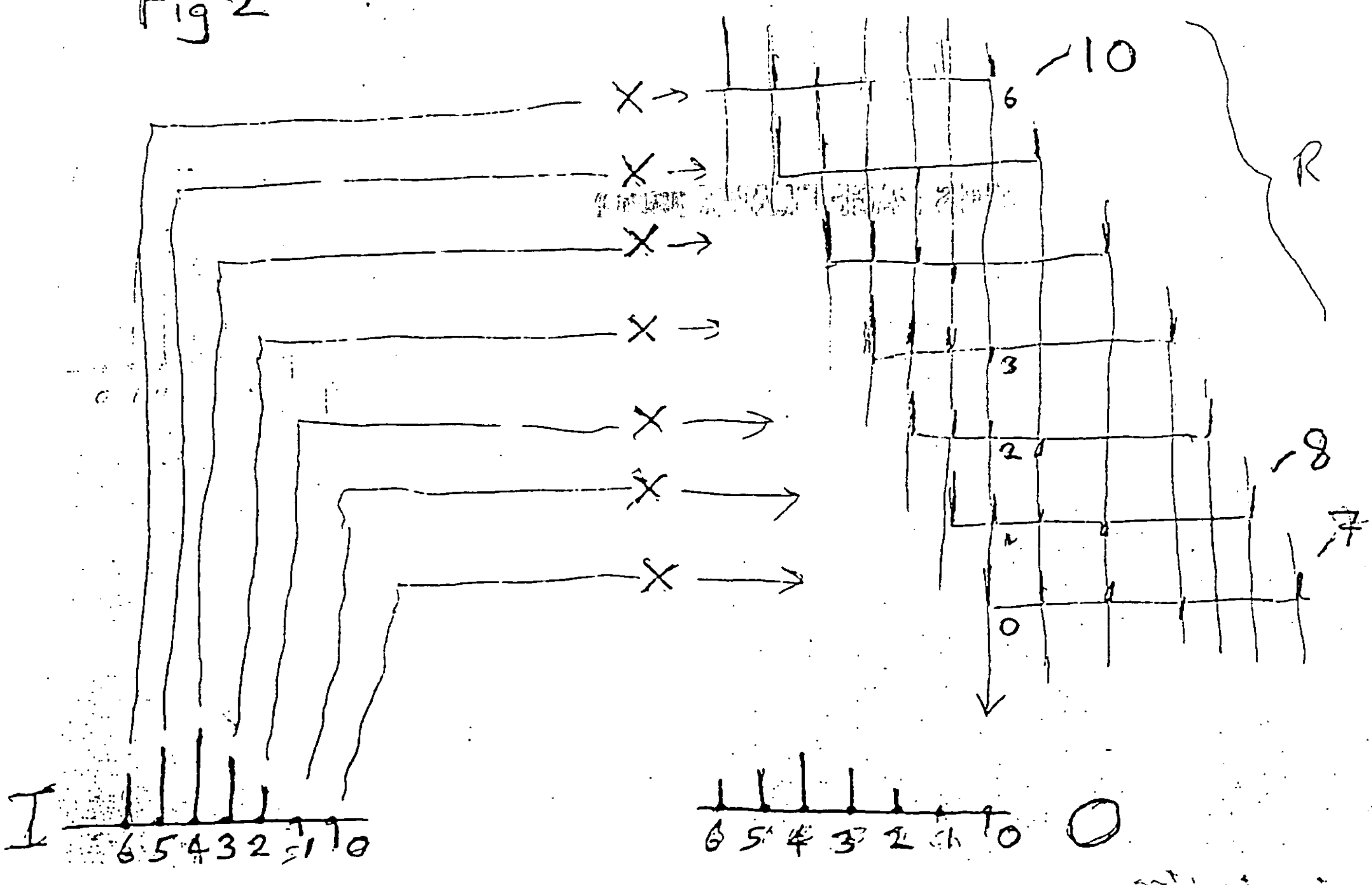

effect unit
D/A
A/D
1
2
3
4
5
0
T
0 1 2 3 4 5 6
R

Fig 2


10
R
6
3
2
8
1
7
0
I
6 5 4 3 2 1 0
6 5 4 3 2 1 0
O

HIS PAGE BLANK (USPTO)

Fig 3a

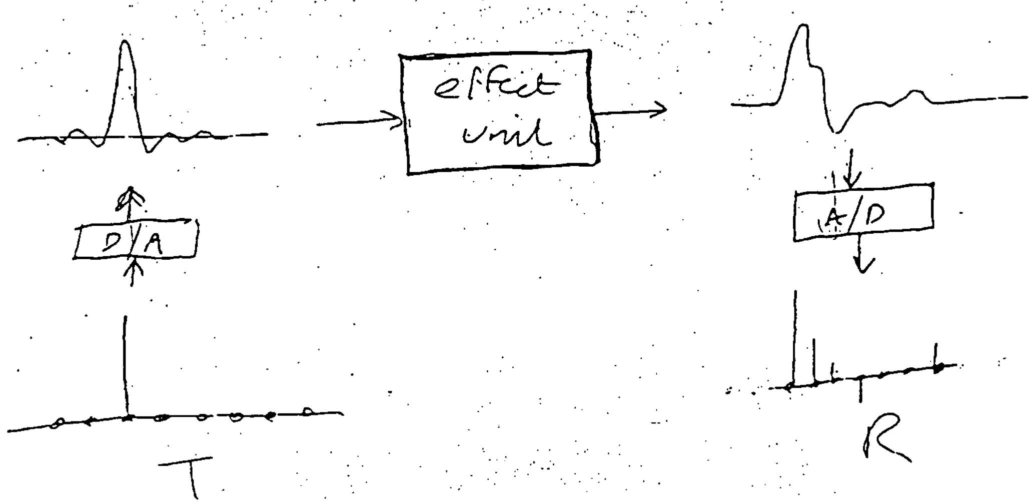
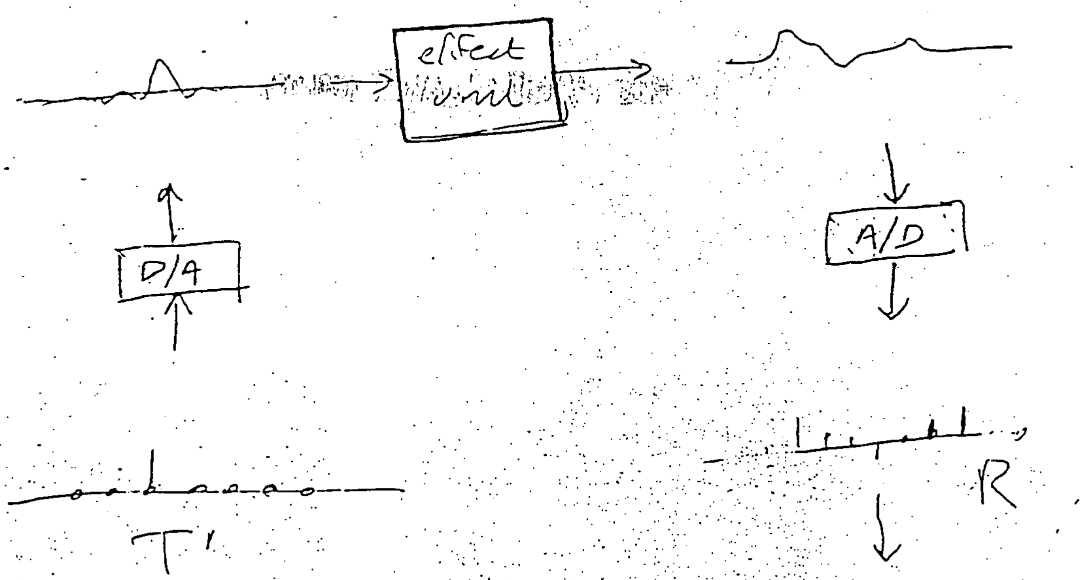


Fig 3b



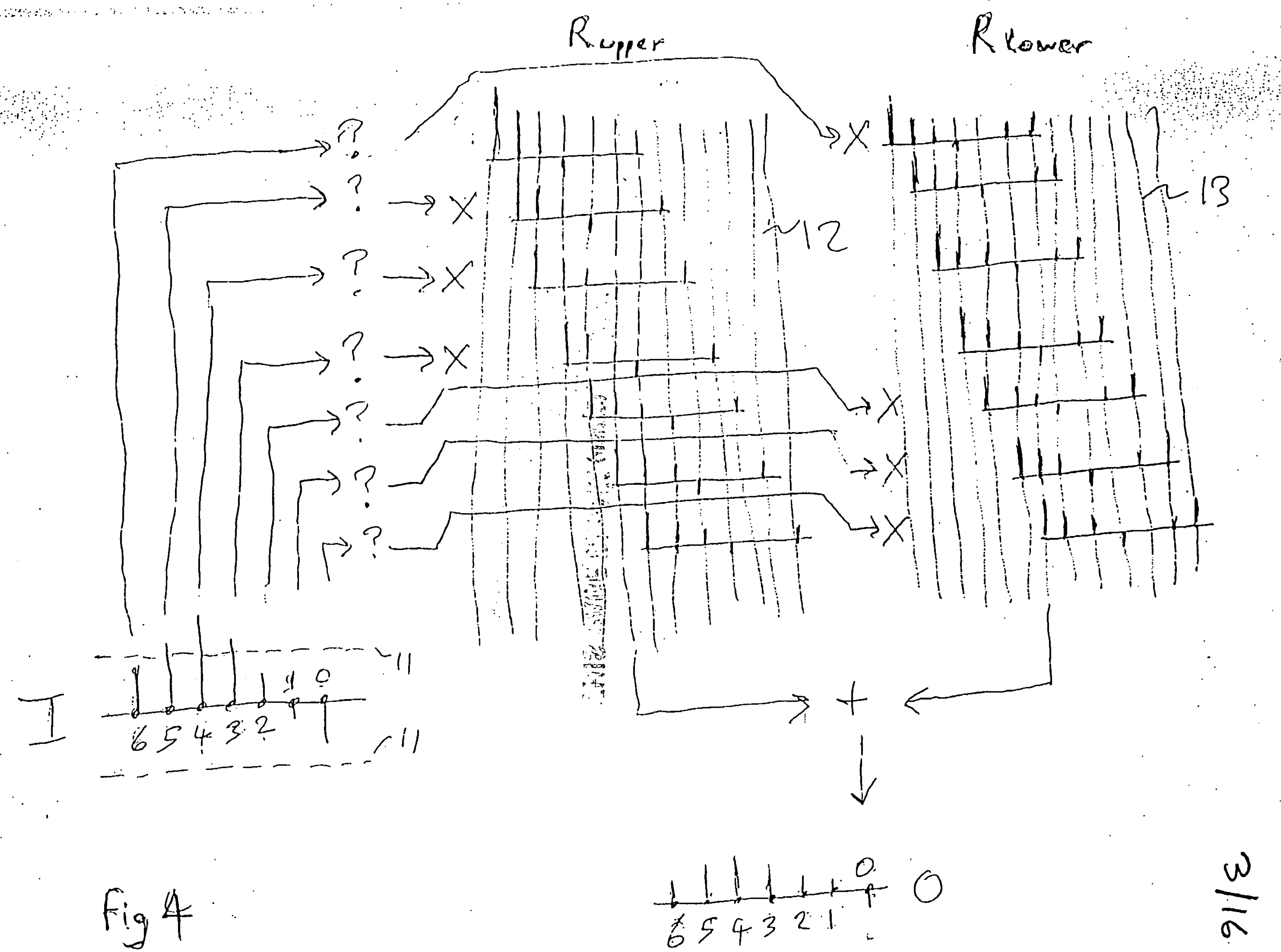


Fig 4

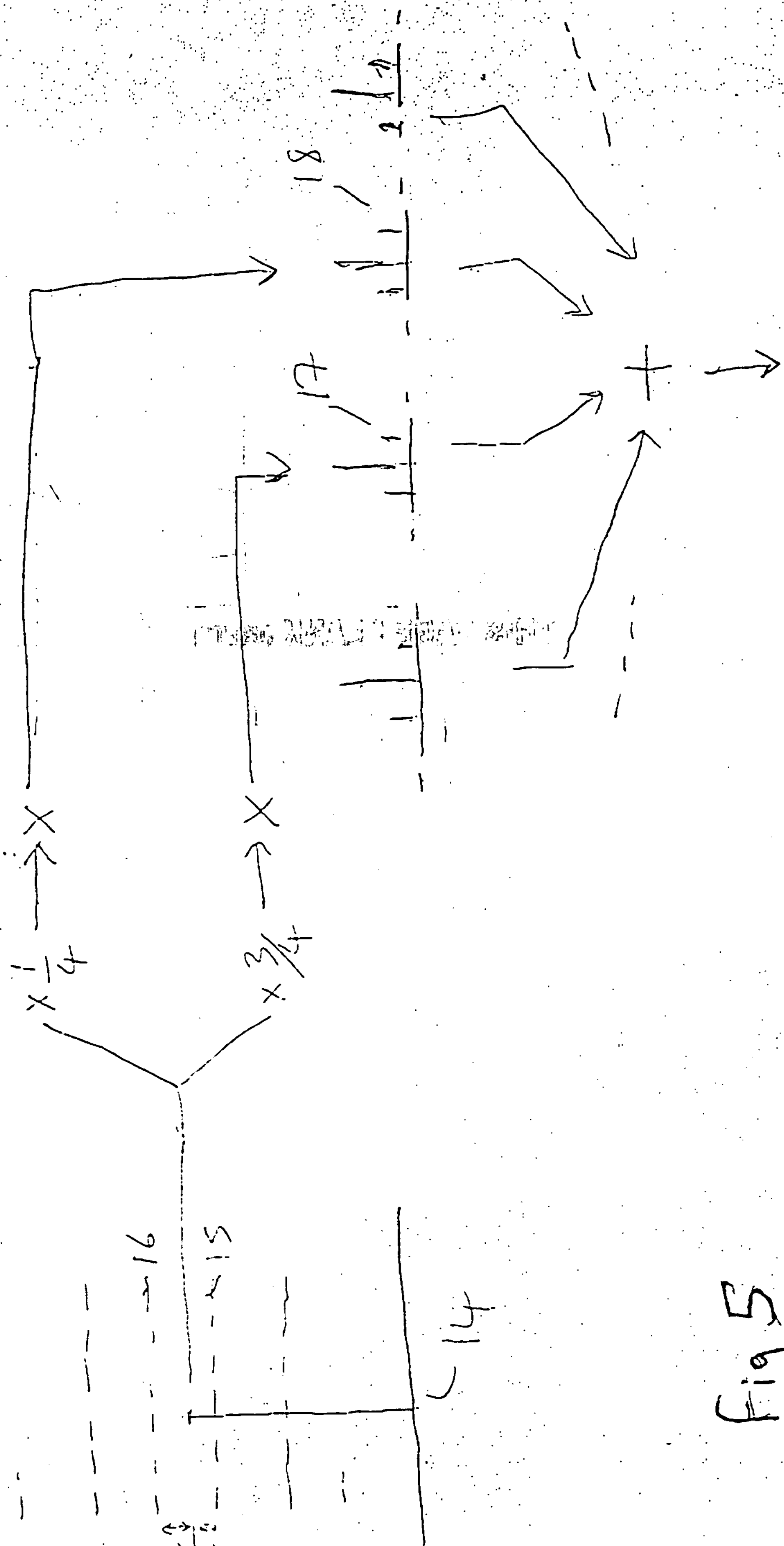


Fig 5

6

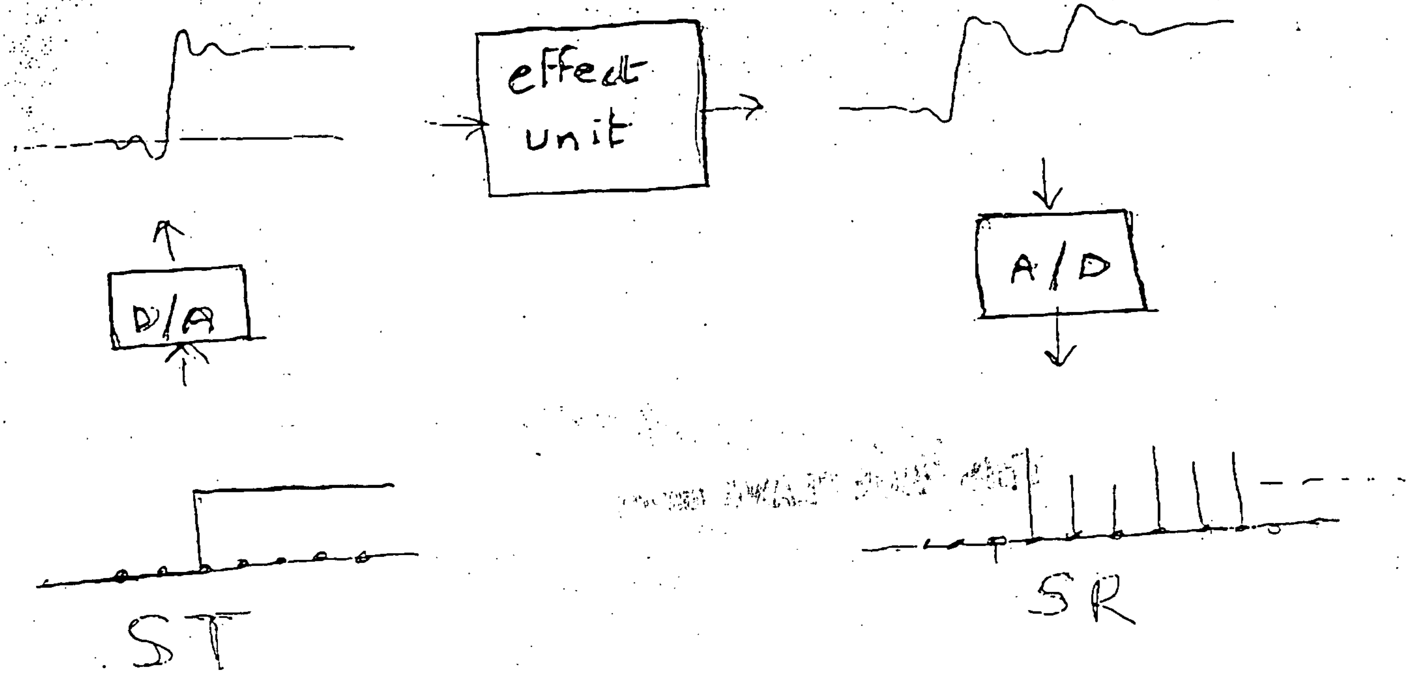
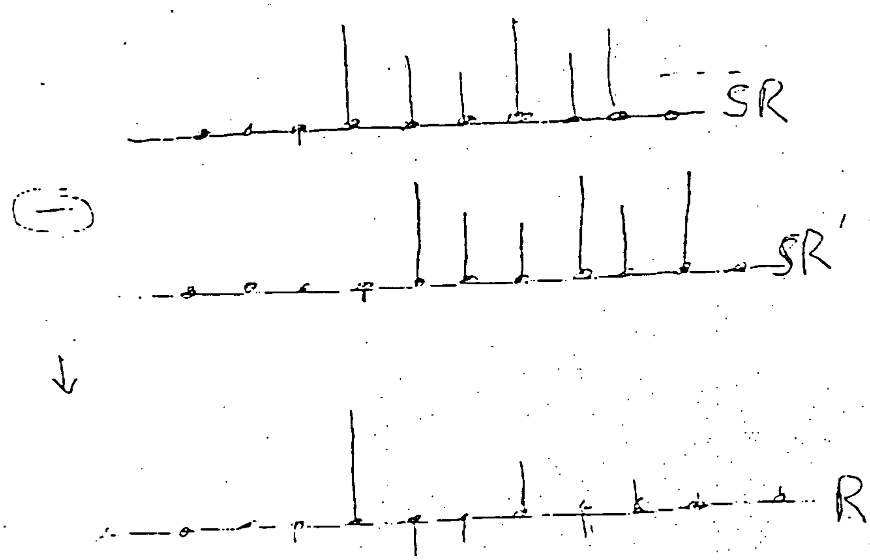
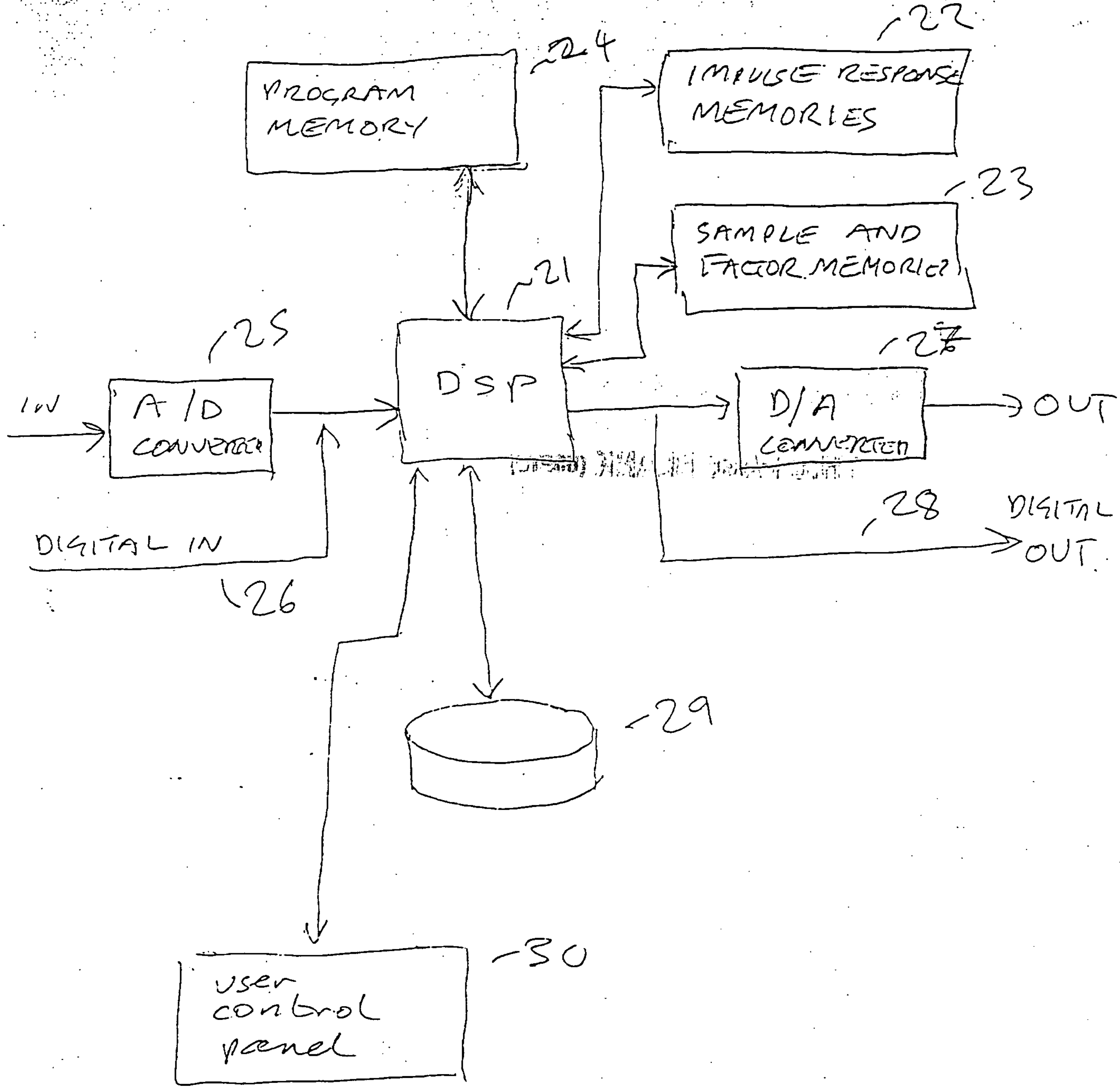


Fig 7



F 8


PROGRAM MEMORY
24
IMPULSE RESPONSE MEMORIES
22
SAMPLE AND FACTOR MEMORIES
23
DSP
21
IN
A/D CONVERTER
25
DIGITAL IN
26
D/A CONVERTER
27
OUT
28
DIGITAL OUT
29
user control panel
30

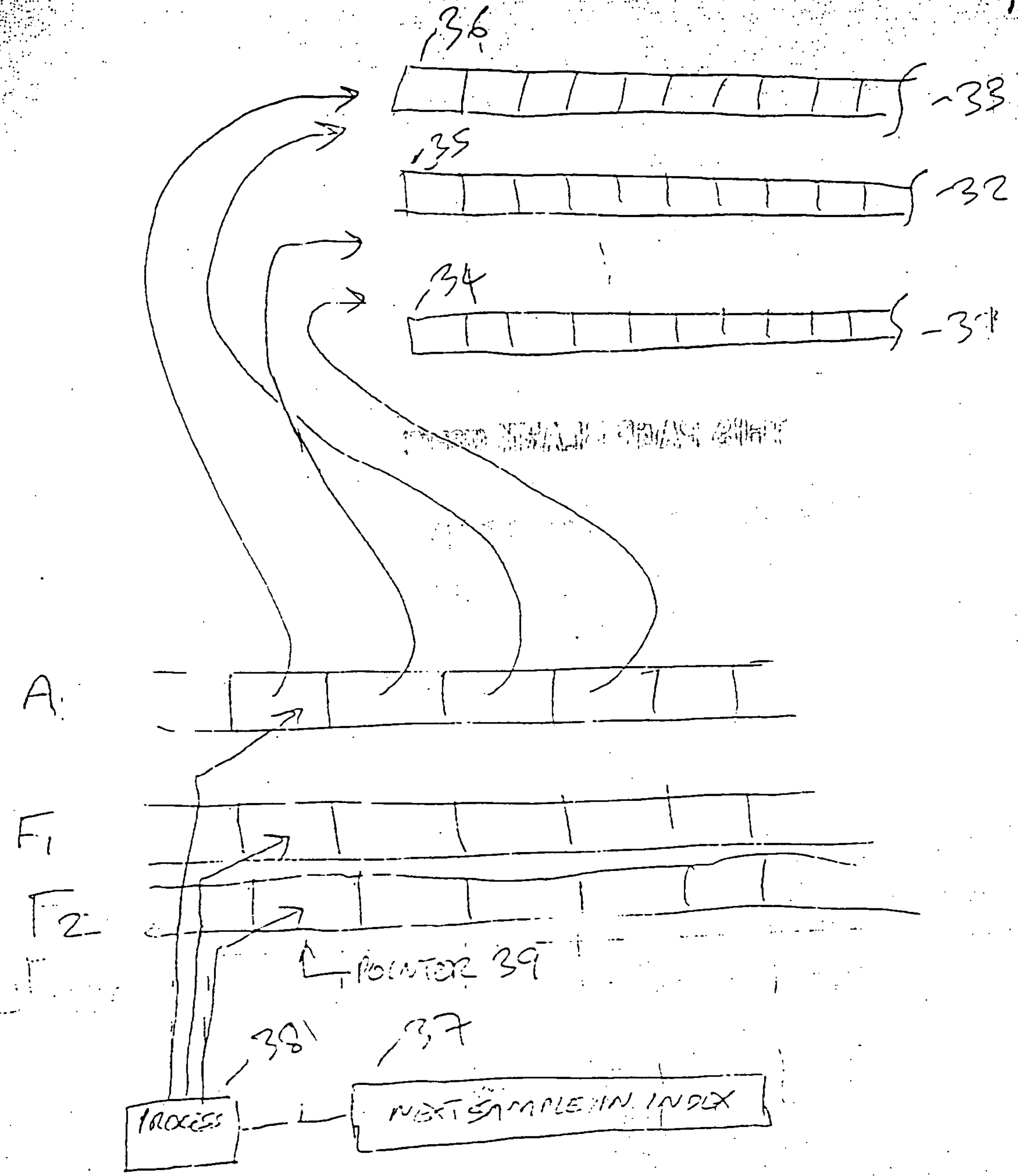
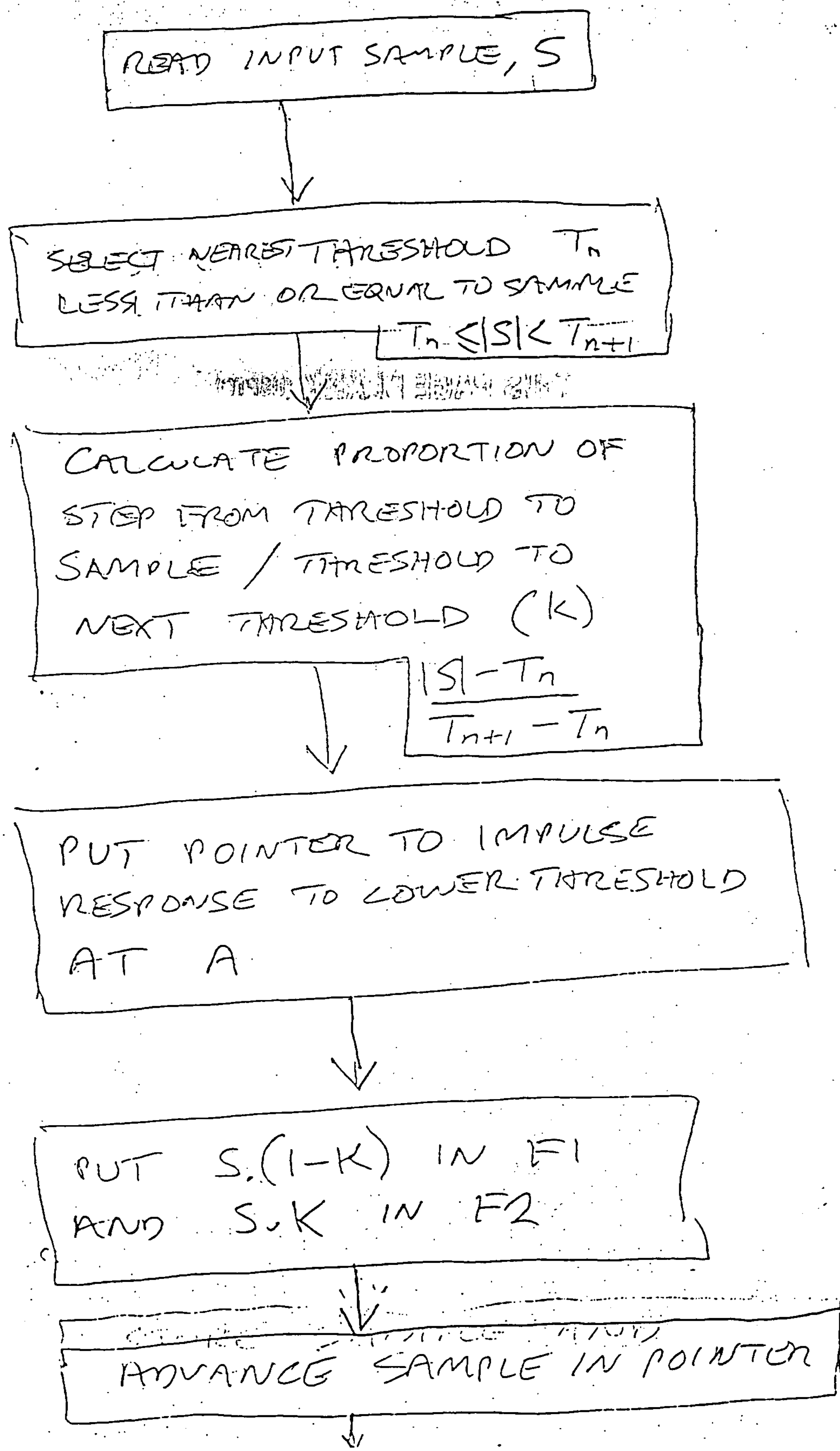


Fig 9 memory storage

Fig 10


READ INPUT SAMPLE, S
SELECT NEAREST THRESHOLD T_n LESS THAN OR EQUAL TO SAMPLE
$T_n \leq |S| < T_{n+1}$
CALCULATE PROPORTION OF STEP FROM THRESHOLD TO SAMPLE / THRESHOLD TO NEXT THRESHOLD (K)
$\frac{|S| - T_n}{T_{n+1} - T_n}$
PUT POINTER TO IMPULSE RESPONSE TO LOWER THRESHOLD AT A
PUT S.(1-K) IN F1 AND S.K IN F2
ADVANCE SAMPLE IN POINTER

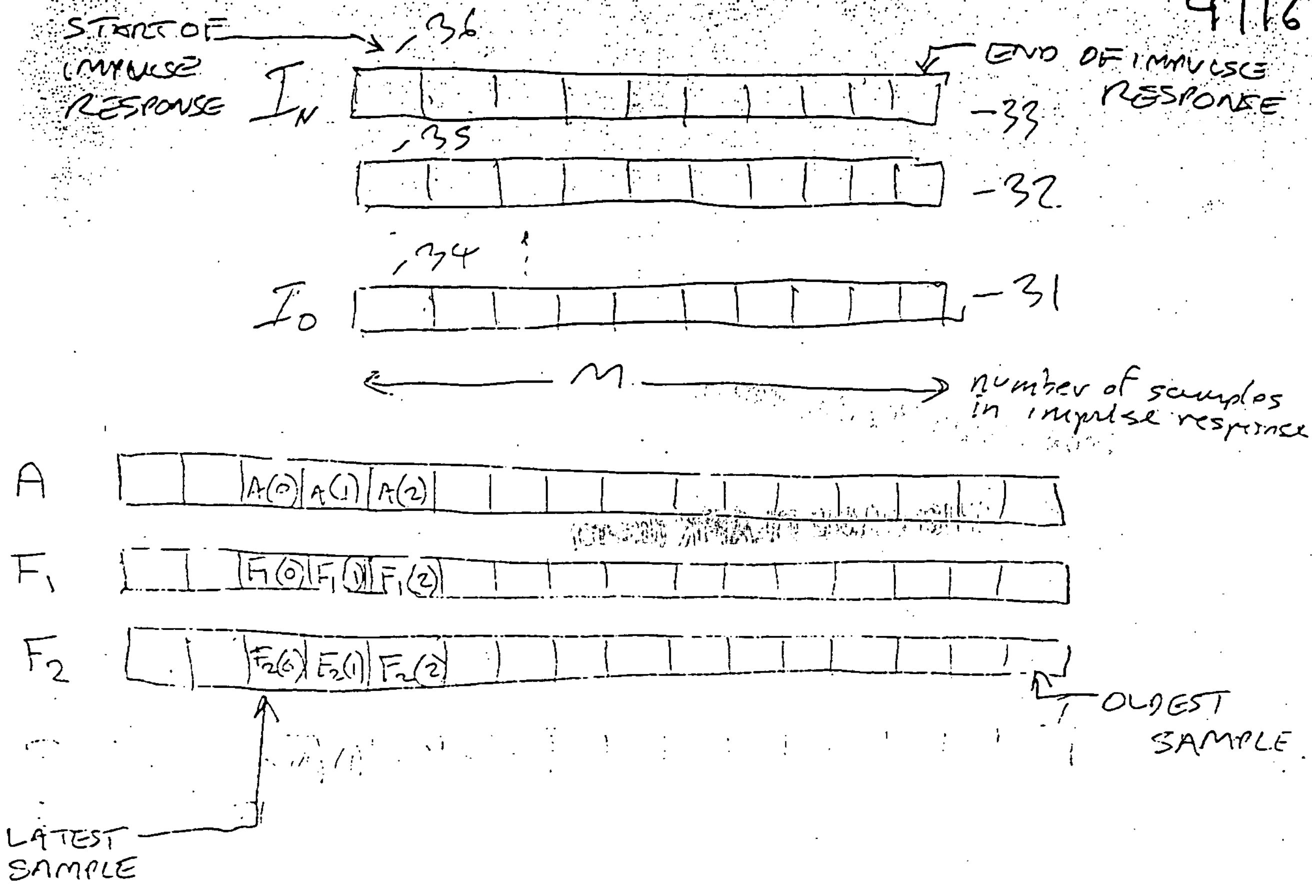

START OF IMPULSE RESPONSE
36
END OF IMPULSE RESPONSE
I_N
33
35
32
34
I_0
31
M
number of samples in impulse response
A
A(0) A(1) A(2)
F_1
F_1(0) F_1(1) F_1(2)
F_2
F_2(0) F_2(1) F_2(2)
OLDEST SAMPLE
LATEST SAMPLE

Fig 11

Fig 2

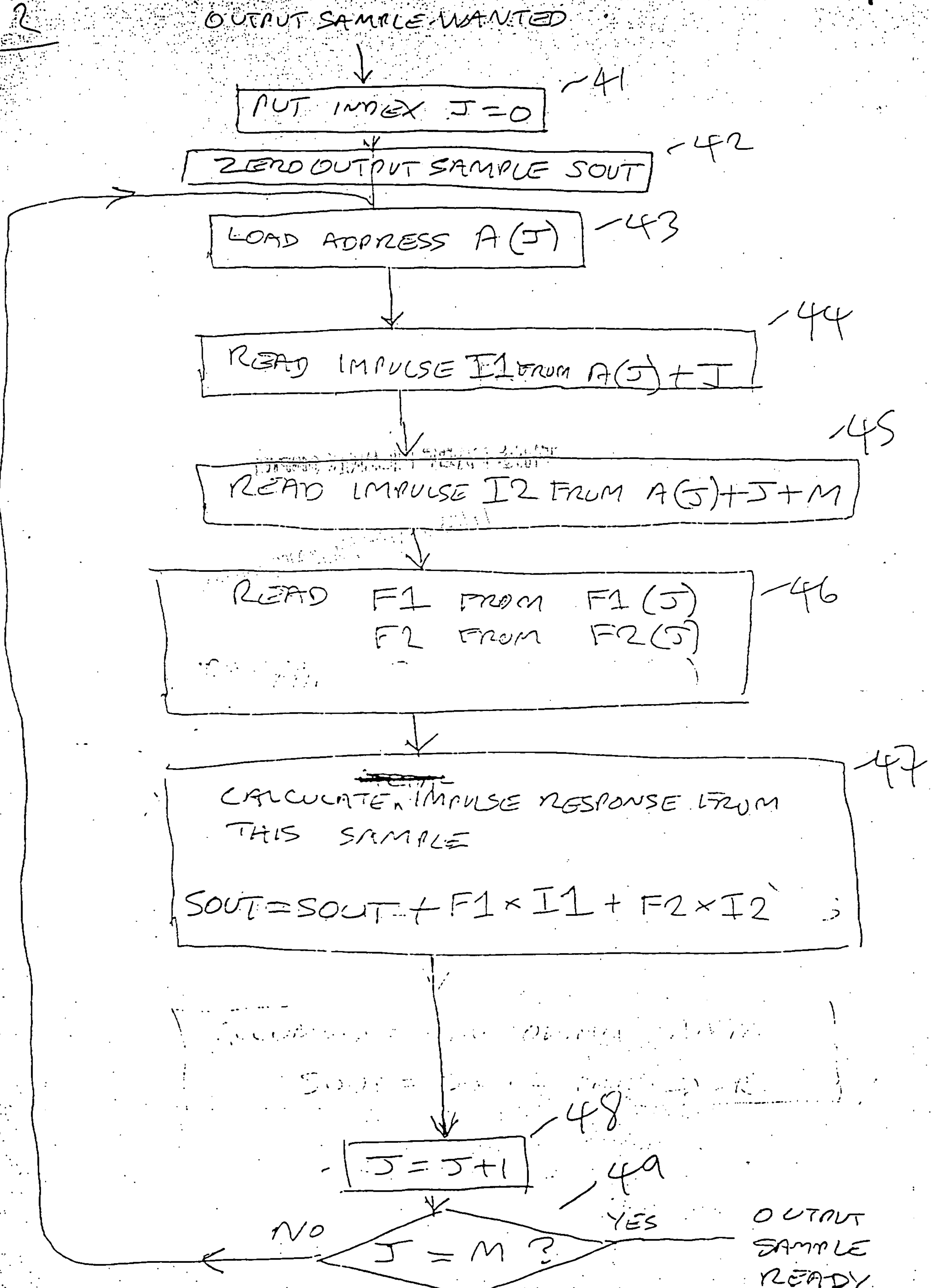

OUTPUT SAMPLE WANTED
PUT INDEX J=0
41
ZERO OUTPUT SAMPLE SOUT
42
LOAD ADDRESS A(J)
43
READ IMPULSE I1 FROM A(J)+J
44
READ IMPULSE I2 FROM A(J)+J+M
45
READ F1 FROM F1(J)
F2 FROM F2(J)
46
CALCULATE IMPULSE RESPONSE FROM THIS SAMPLE
SOUT=SOUT+F1×I1+F2×I2
47
J=J+1
48
J=M?
49
NO
YES
OUTPUT SAMPLE READY

Fig 13

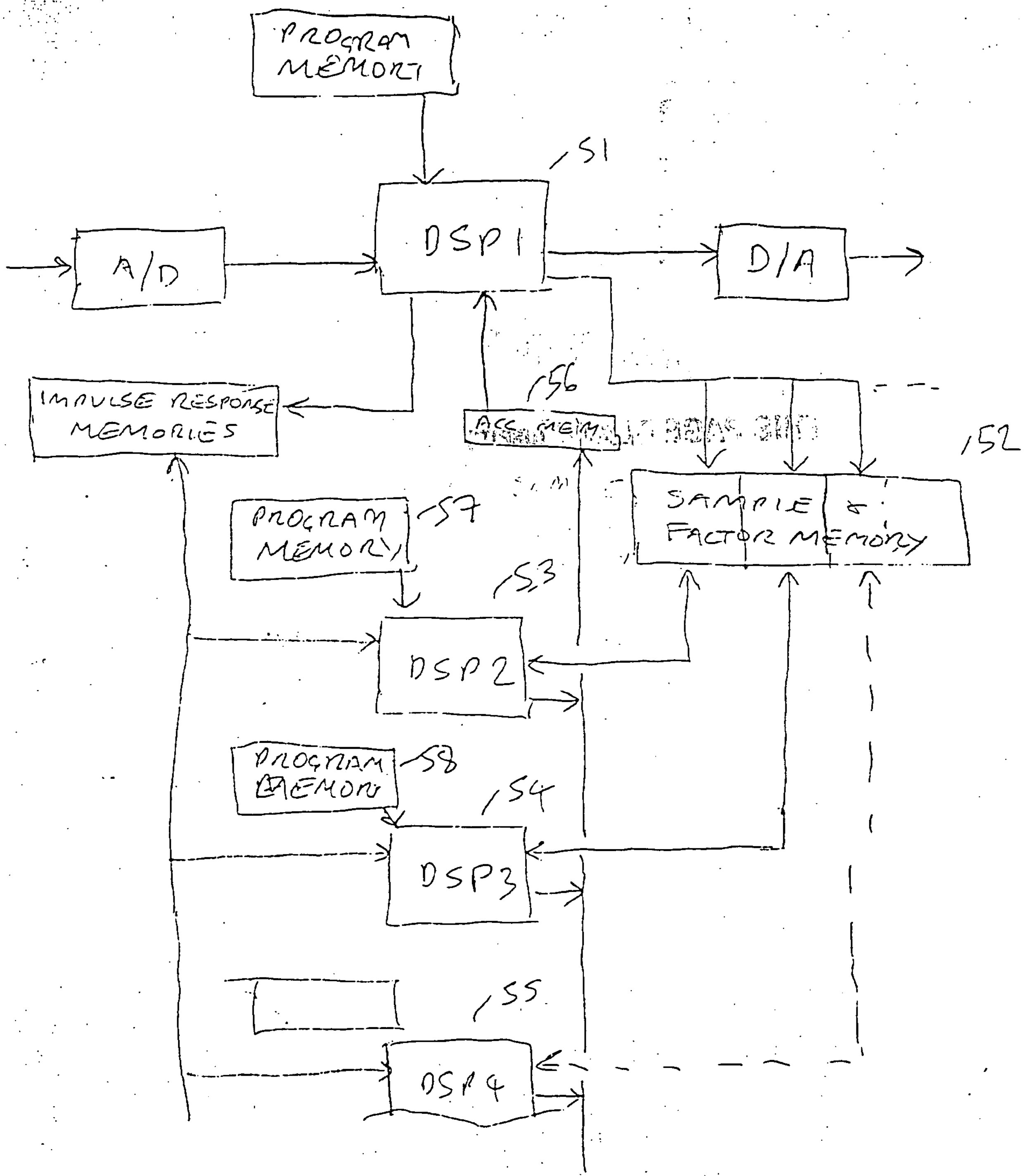

PROGRAM MEMORY
51
DSP1
A/D
D/A
IMPULSE RESPONSE MEMORIES
56
ACC. MEM.
52
SAMPLE & FACTOR MEMORY
PROGRAM MEMORY
57
53
DSP2
PROGRAM MEMORY
58
54
DSP3
55
DSP4

Fig 14

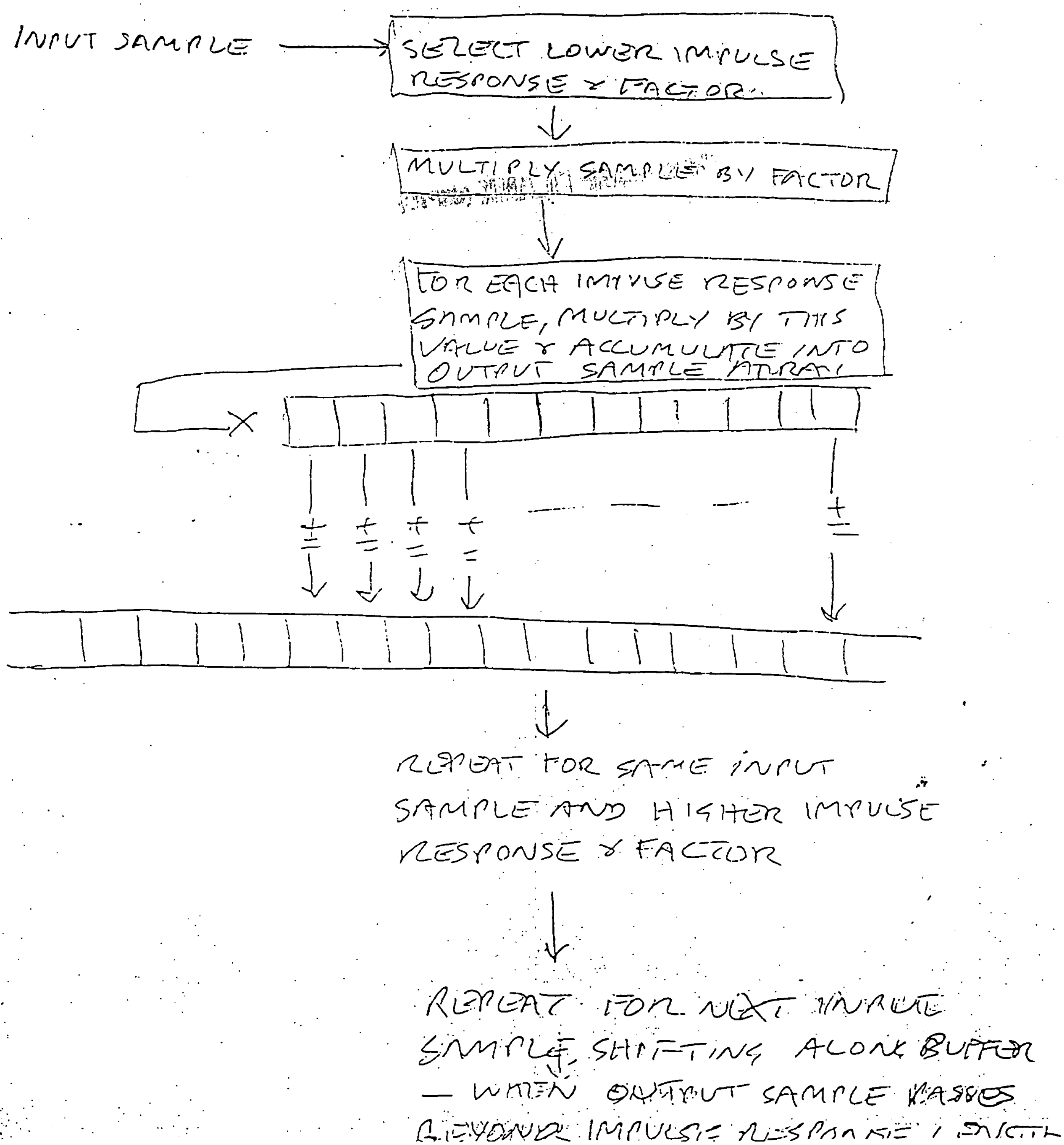
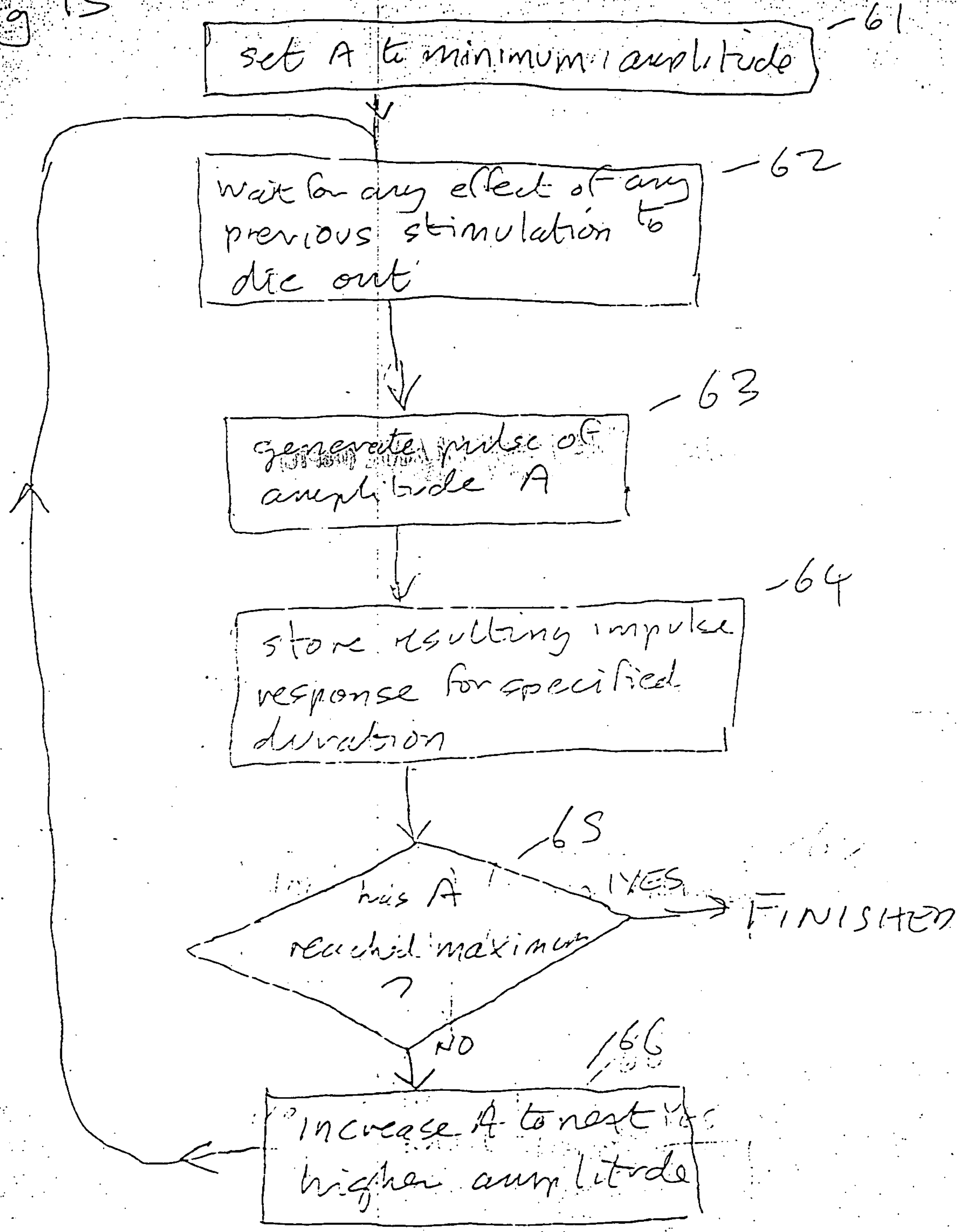

INPUT SAMPLE
SELECT LOWER IMPULSE RESPONSE & FACTOR.
MULTIPLY SAMPLE BY FACTOR
FOR EACH IMPULSE RESPONSE SAMPLE, MULTIPLY BY THIS VALUE & ACCUMULATE INTO OUTPUT SAMPLE ARRAY
X
+=
REPEAT FOR SAME INPUT SAMPLE AND HIGHER IMPULSE RESPONSE & FACTOR
REPEAT FOR NEXT INPUT SAMPLE SHIFTING ALONG BUFFER — WHEN OUTPUT SAMPLE PASSES BEYOND IMPULSE RESPONSE LENGTH

fig 15


set A to minimum amplitude
61
wait for any effect of any previous stimulation to die out
62
generate pulse of amplitude A
63
store resulting impulse response for specified duration
64
has A reached maximum ?
65
YES
FINISHED
NO
increase A to next higher amplitude
66

THIS PAGE BLANK (USPTO)

16

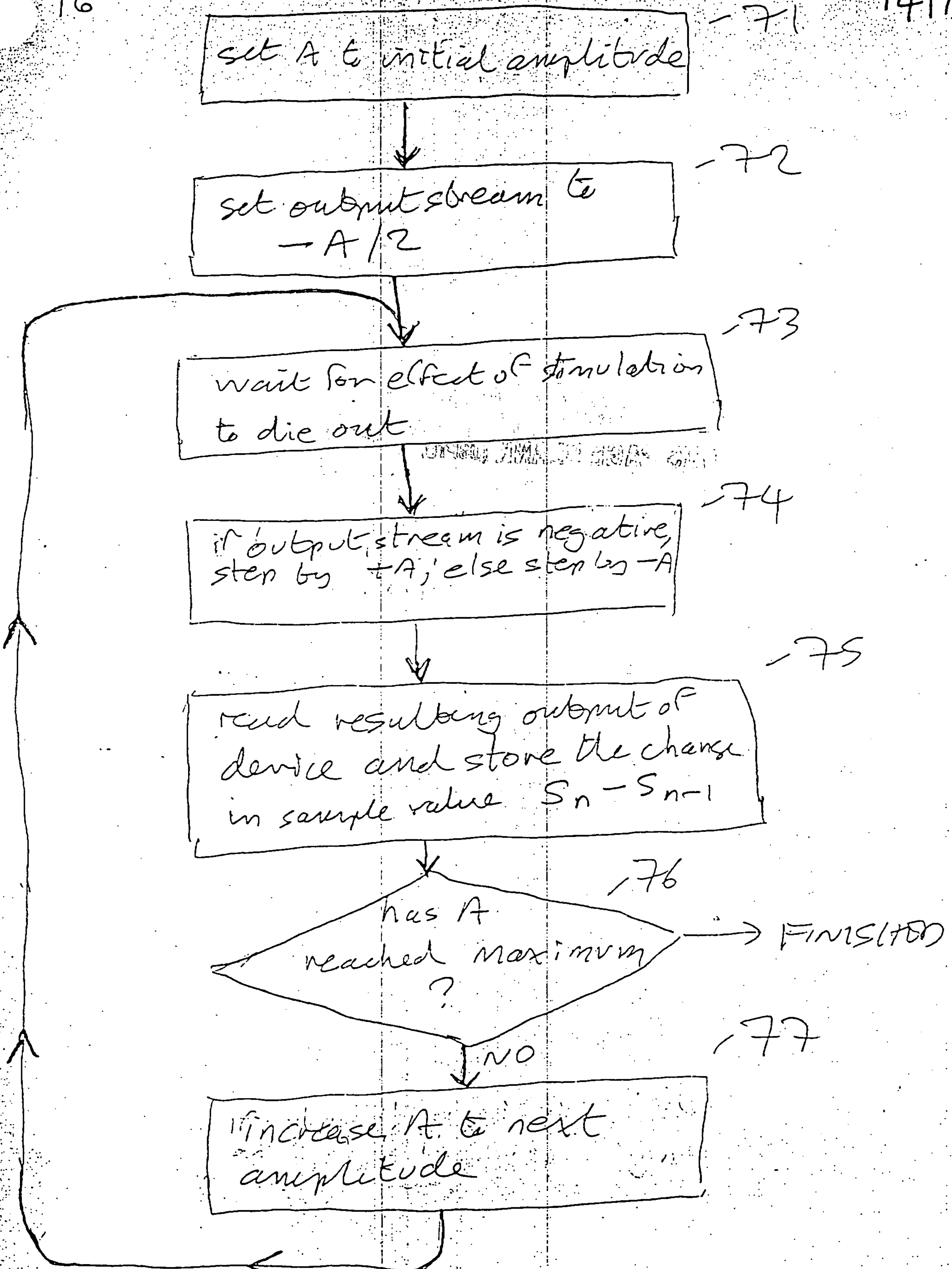

set A to initial amplitude
71
set output stream to −A/2
72
wait for effect of stimulation to die out
73
if output stream is negative, step by +A; else step by −A
74
read resulting output of device and store the change in sample value $S_n - S_{n-1}$
75
has A reached maximum ?
76
FINISHED
NO
increase A to next amplitude
77

Fig 17

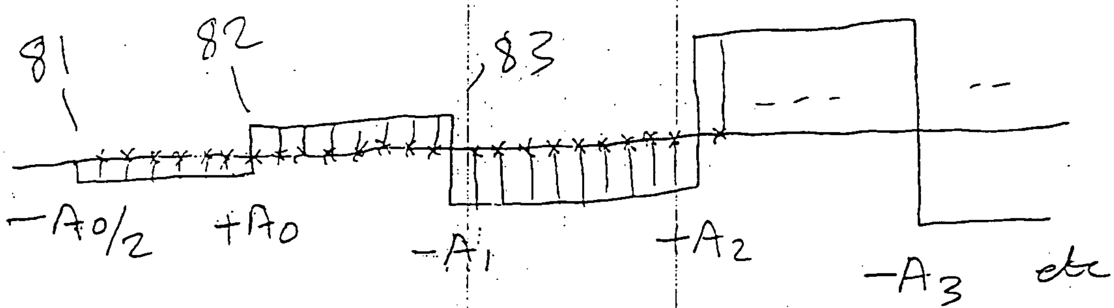
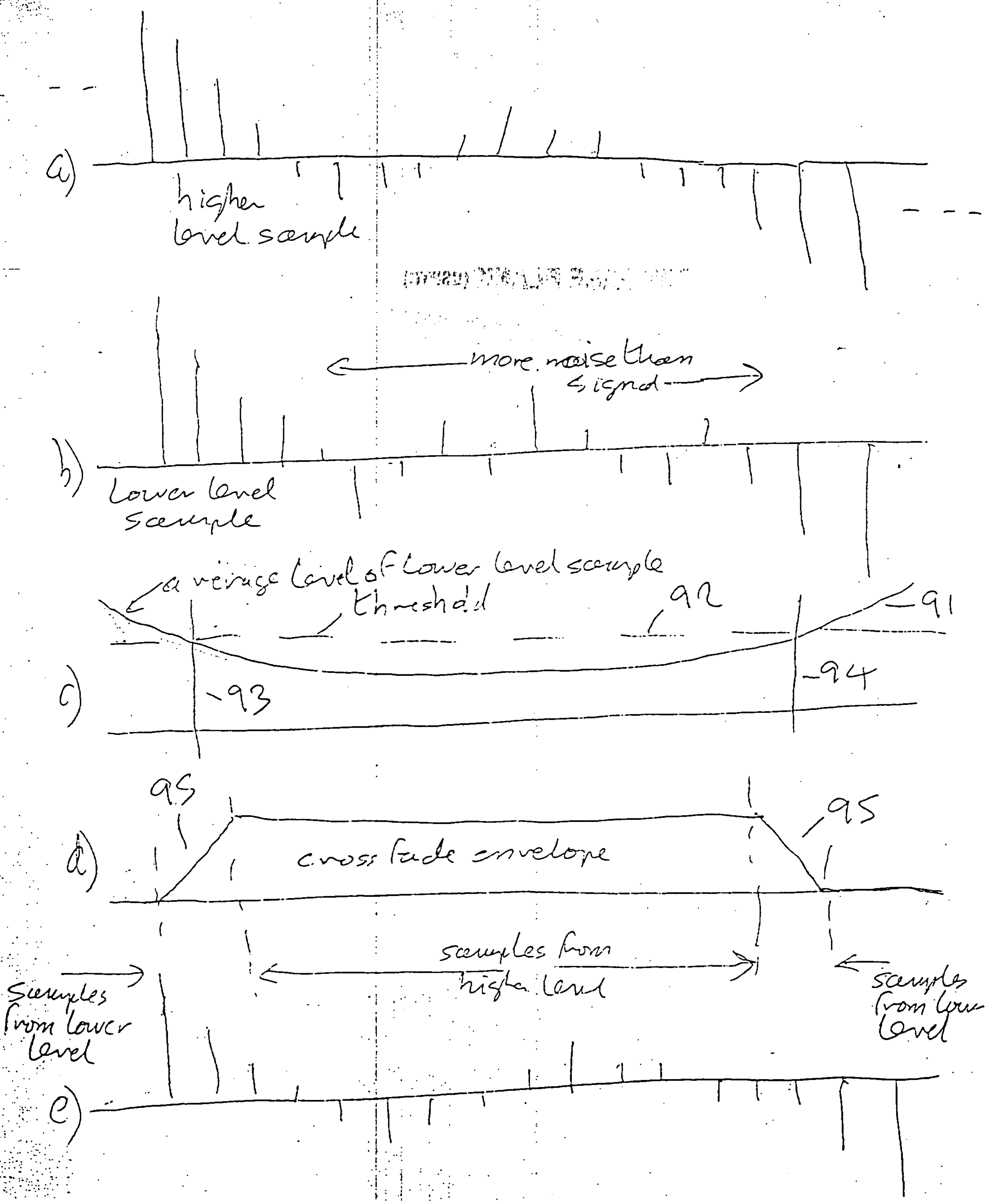
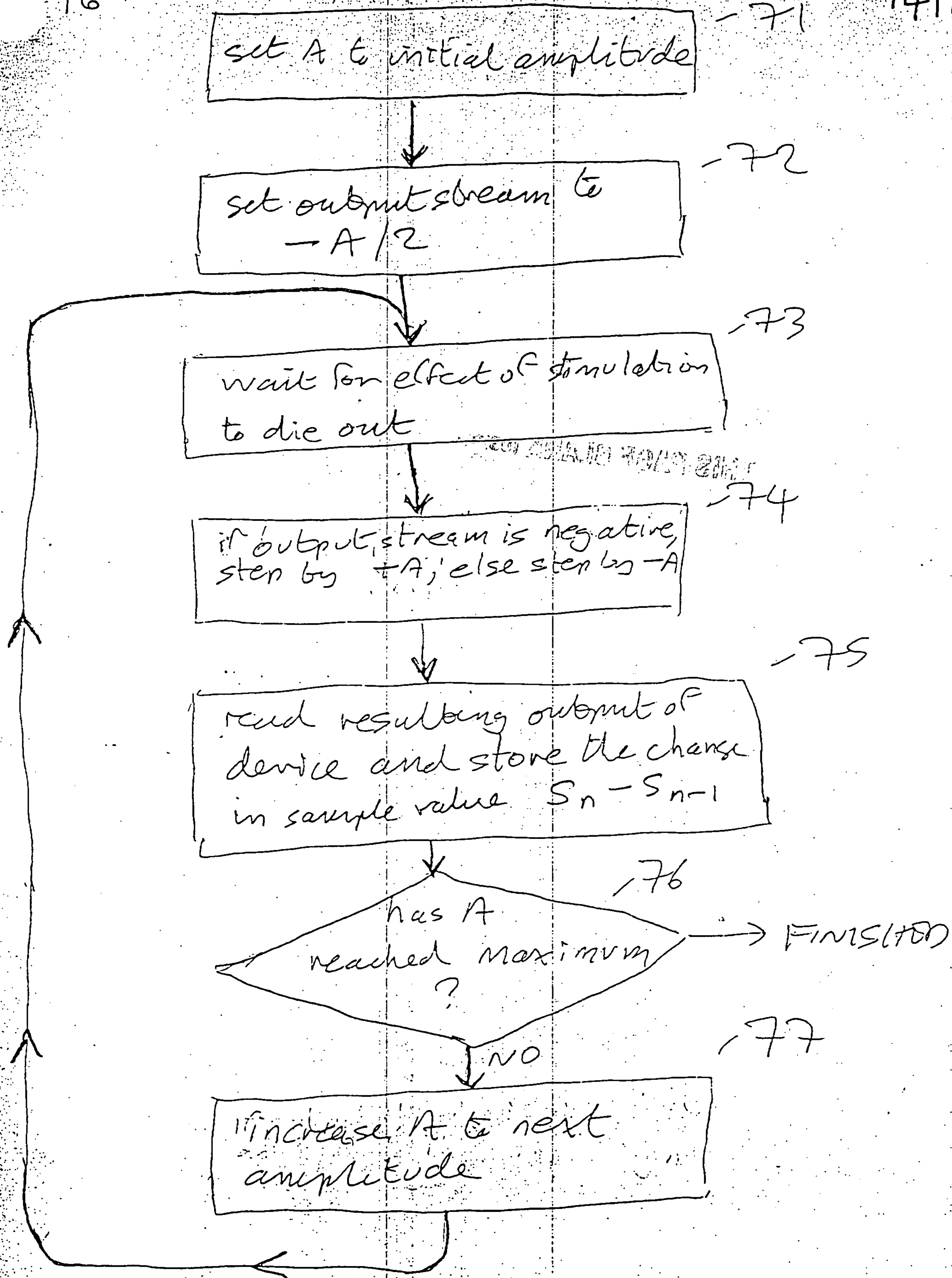


Fig 18

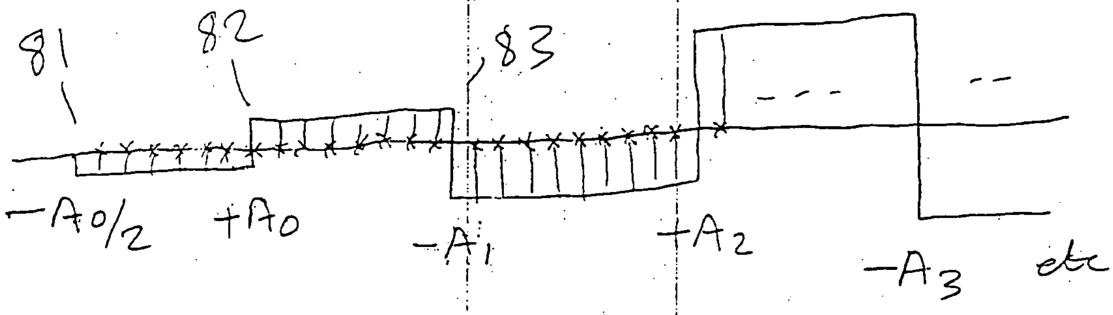

a)
higher
level sample
b)
more noise than
signal
Lower level
sample
c)
average level of lower level sample
threshold
92
91
93
94
d)
95
95
cross fade envelope
samples from
higher level
samples
from lower
level
samples
from lower
level
e)

F 16


71
set A to initial amplitude
72
set output stream to −A/2
73
wait for effect of stimulation to die out
74
if output stream is negative, step by +A; else step by −A
75
read resulting output of device and store the change in sample value Sn − Sn-1
76
has A reached maximum ?
FINISHED
NO
77
increase A to next amplitude

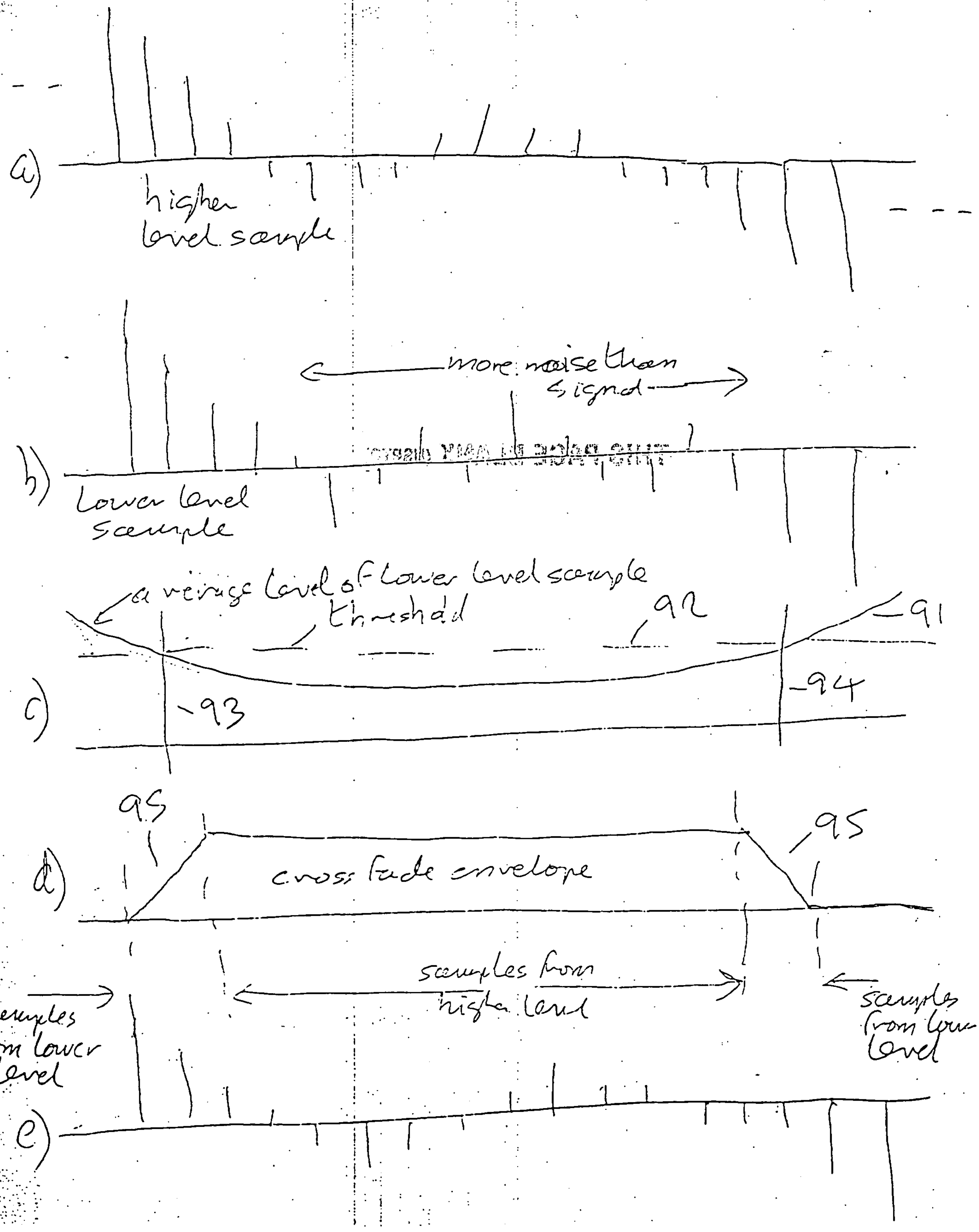
THIS PAGE BLANK (USPTO)

Fig 17



THIS PAGE BLANK (USPTO)

Fig. 18


a)
higher level sample
b)
more noise than signal
Lower level sample
average level of lower level sample
threshold
92
91
c)
93
94
95
d)
cross fade envelope
95
samples from higher level
samples from lower level
samples from lower level
e)

PCT/GB97/02259
5.1.77
Reddie - Grose

THIS PAGE BLANK (USPTO)